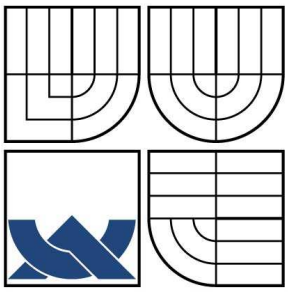
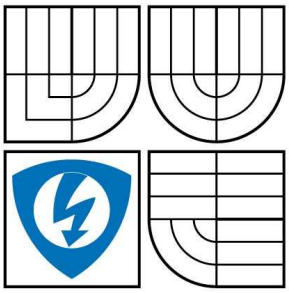


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VÝVOJ APLIKACE PRO MODELOVÁNÍ FYZIKÁLNÍCH JEVŮ

DEVELOPMENT OF APPLICATION FOR SIMULATION OF PHYSICALS EFFECTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Milan Polách
AUTHOR

VEDOUCÍ PRÁCE — Ing. Vladimír Holcman, Ph.D.
SUPERVISOR

BRNO 2009

VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Milan Polách	***ID:***	98121
Ročník:	3	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

Vývoj aplikace pro modelování fyzikálních jevů

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvoření nového prostředí pro výuku a simulaci základních fyzikálních jevů. Cíle celé práce lze rozdělit do následujících kapitol:
- vytvoření uživatelského rozhraní,
- analýza vstupních údajů do animačního prostředí,
- vytvoření animací vybraných fyzikálních úloh,
- kompletace všech dílčích částí do jednoho jednoduché celku.

DOPORUČENÁ LITERATURA:

[1] Halliday, Resnick, Walker: Fyzika, Prometheus a Vutium, Praha 2000, 2005. Český překlad: Dub,P. a kol.
[2] Uhdeová a kol. Fyzikální praktikum, Ústav fyziky FEKT VUT v Brně, Brno 2004, 2005, 2006. Skriptum potřebné pro laboratorní cvičení.
[3] P.Pokorný, Blender - naučte se 3D grafiku, BEN 2006.

Termín zadání:	9.2.2009	***Termín odevzdání:***	2.6.2009

Vedoucí práce: Ing. Vladimír Holcman, Ph.D.

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Abstrakt

Cílem této bakalářské práce bylo vytvořit výukový systém pro efektivnější výuku fyzikálních jevů a nahrazení starších používaných systémů. Celek se skládá z několika částí, kde na začátku jsou teoretické informace zobrazeny na internetových stránkách, dále je možné spustit simulaci fyzikálních jevů a na závěr je možné si na animaci prohlédnout, jak daný jev vypadá v reálném prostředí při reálných situacích. Animace pro simulace jevů v reálném prostředí jsou tvořeny programem Blender a pomocí programovacího jazyka Python.

Klíčová slova

Blender, simulace fyzikálních jevů, Python,e-learning,animace

Abstrakt

The aim of this bachelor's thesis was to create an educational system for more effective teaching physical phenomena and the replacement of older systems used. Whole project is composed of several parts. At the begining, the theoretical information are shown on website. Next is a simulation of physical phenomena and in conclusion, how the phenomena looks like in a real situation is presented. Animations for simulations are created in Blender, using programming language Phyton.

Keywords

Blender, simulation of physicals effects, Python, e-learning, animation

POLÁCH, Milan. VÝVOJ APLIKACE PRO MODELOVÁNÍ FYZIKÁLNÍCH JEVŮ . [s.l.], 2009. 37 s. , 1 CD-ROM. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí bakalářské práce Ing. Vladimír Holcman, Ph.D.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Vývoj aplikace pro modelování fyzikálních jevů jsem vypracoval samostatně pod vedením vedoucího bakalářské práce s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Vladimíru Holcmanovi, Ph.D., za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

V Brně dne . . . .

(podpis autora)

Úvod

Cílem bakalářské práce je vytvořit nové prostředí pro výuku fyzikálních jevů. Základ by měly tvořit internetové stránky, kde se student seznámí s fyzikálním jevem, zjistí, kde se s ním může setkat a bude si moci spustit jednoduchou animaci, kde uvidí probíraný děj. Pokud bude mít dále zájem, bude si moci stáhnout a spustit program, kde po zadání počátečních hodnot bude moci spustit animaci na vykreslení trajektorie. Druhá možnost na spuštění bude zanimování děje pomocí programu Blender. Tento program je spíše zaměřen na ukázání děje v praxi, než vysvětlení teorie děje.

Moderní postupy výuky se zaměřují na co nejefektivnější, ale zároveň na co nejsrozumitelnější způsob předání informace. Pokud někdo potřebuje získat informace, stačí se pouze připojit na internet a informace vyhledat. Na internetu se dozví informace, které hledal, ale nalezne zde i mnoho dalšího např. praktické ukázky. Tomuto by měla napomoci i tato práce.

Bakalářský projekt vychází z požadavků lektorů fyziky na komplexnější a názornější systémy pro počítačovou výuku a e-learning. Celek by měl nahradit stávající program Fámulus, u kterého jsou problémy s rychlostí zobrazovaného děje. Fámulus byl koncipován na méně výkonné počítače a z tohoto důvodu se musel ručně upravovat zdrojový kód, aby byla animace přehledná.

Pro animaci bude použit OpenSource program Blender. Tento program je sice primárně určen na 3D modelování, ale má i GameEngine, již název napovídá, je zaměřený na hry. Z GameEngine využijeme jen malou část a to na samotné spuštění animace. Nejprve se pomocí Python skriptu načtou počáteční hodnoty vytvořené programem na zobrazení trajektorie, následně se provede vypočítání dráhy objektu. Po spuštění GameEngine v Blenderu bude stačit zmačknout pouze několik kláves na spuštění animace.

V bakalářské práci by měly být vytvořeny simultánní animace pro fyzikální jevy a to vrh vzhůru, šikmý vrh, volný pád a vodorovný vrh. Tyto animace studentům v co největší míře přiblíží probíraný fyzikální jev.

Program na vytvoření počátečních hodnot a zobrazení trajektorie je náplní jiné bakalářské práce.

1 Teoretický úvod

1.1 Historie Blenderu

Blender vznikl v roce 1996 jako modelovací nástroj animačního studia NeoGeo. V roce 1998 byl Blender zveřejněn ve verzi 1.25 a podporoval operační systémy FreeBSD a Linux. Následně byla založena společnost NaN(Not a Numer), která pracovala na vývoji Blenderu Creator. [2]

V roce 1999 byla uvolněna verze pro Windows. Byly přidány nové modelační funkce, které se daly dokoupit za 95 USD. Ve verzi 2.25 vznikla komerční verze Blenderu, Blender Publisher. Tato verze byla k dispozici za 11000 Kč. Bohužel vývoj nešel tak, jak si firma NaN představovala a 12. března 2002 vyhlásila bankrot. [2]

Firma NaN odmítla zveřejnit zdrojové kódy Blenderu. To se povedlo až po vyjednávání Tona Roosendaala za následujících podmínek :

- -vznikne nekomerční asociace Blender Foundation
- -nadace zaplatí společnosti poplatek 100 000 USD
- -nadace znovu zpřístupní dokumentaci a webové stránky Blenderu

Několik zastánců Blenderu a bývalých zaměstnanců začalo shánět potřebné finance na odkoupení Blenderu. Minimální vklad byl 50 euro a tím získal přispěvatel členství v komunitě Blender Foundation.

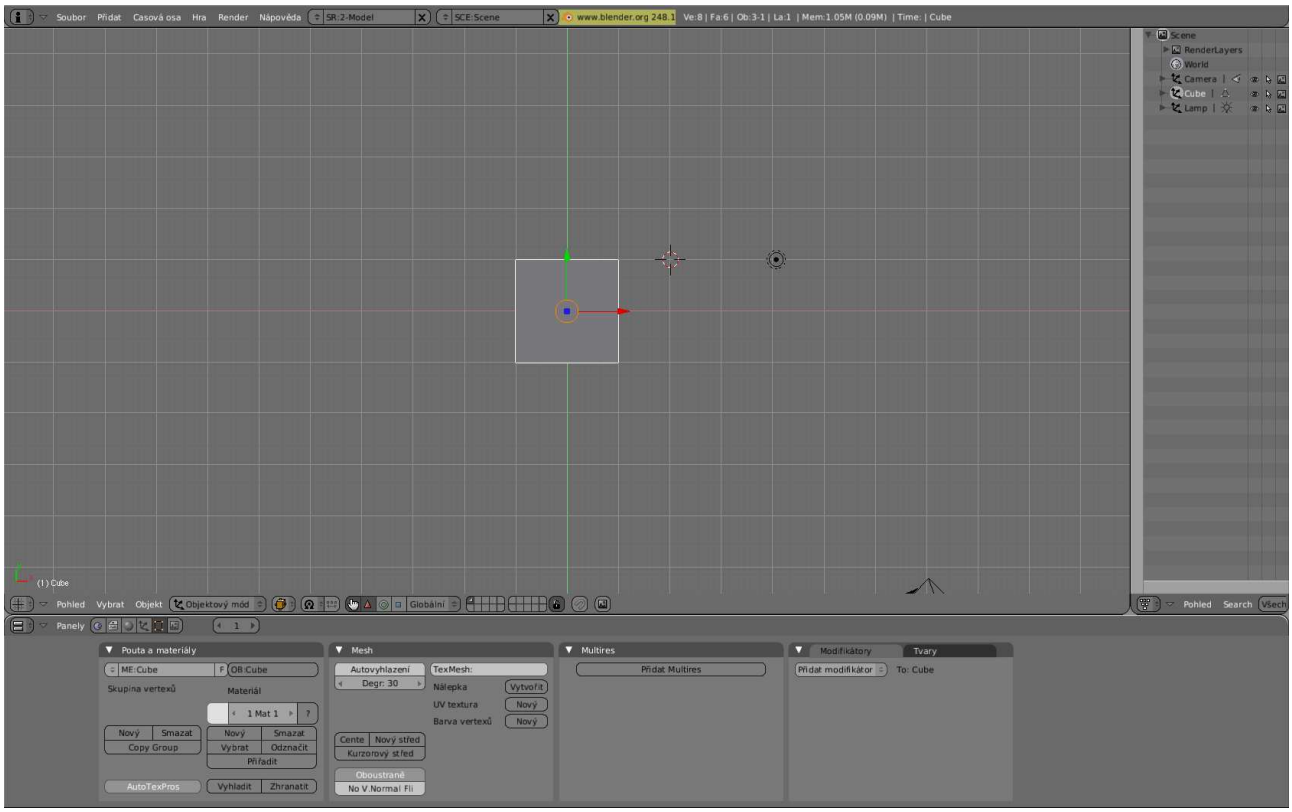
Velkým překvapením bylo, že 100 000 USD dali dohromady za pouhých 7 týdnů a dne 13. října 2002 byl Blender zveřejněn pod licencí GNU GPL a byly spuštěny internetové stránky www.blender.org

Nyní se vývoji Blenderu věnují dobrovolníci z celého světa pod vedením Tona Roosendaala. V roce 2005 založila Blender Foundation projekt „Project Orange“ , který měl za úkol vytvořit animovaný film. První film se jmenoval Elefant Dream. [2]

Díky velkému úspěchu filmu založil Ton Roosendaal Blender Institut, který má za úkol koordinovat a vytvářet OpenSource projekty související s projekty 3D filmu, her a vizuálních efektů.

V dubnu vydal projekt Peach Project otevřený film „Big Buck Bunny“. V současné době se pracuje na hře Apricot. [2]

1.2 Uživatelské rozhraní Blenderu

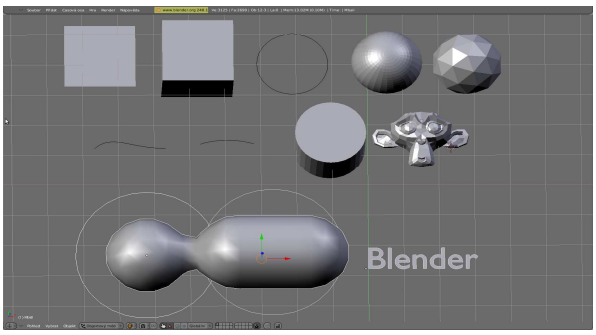


Obr.1.1 Náhled na rozhraní Blenderu

Rozhraní Blenderu je plně přizpůsobitelné. My si zde popíšeme základní nastavení modelovacího rozhraní. Základní rozhraní se skládá z horní uživatelské lišty, 3D okna a ovládacích panelů (Obr.1.1).

1.2.1 Uživatelské nastavení

V horním uživatelském nastavení v menu Soubor se nachází možnosti pro ukládání a načítání souboru *.blend , import a export případně ukládání jako spustitelný exe soubor, pokud pracujeme s GameEngine.



Obr. 1.2 Ukázka základních objektu Blenderu

V menu Přidat máme možnost přidávat do scény křivky, objekty, meta objekty, světla, kamery a opičku Suzanne (Obr. 1.2). Tyto objekty jdou pak dále přizpůsobitelné našim požadavkům.

V menu Časová osa jsou různé možnosti pro ulehčení práce s časem, které částečně využijeme při animaci.

Dále tu máme menu Hra a render. Tato menu jsou důležitá pro vytvoření výsledných obrázků nebo animaci. My využijeme jen částečně možnosti menu Hry, kde se nastavují druhy použitých textur pro Game engine a dále možnost spustit hru. Render nebudeme využívat, protože bude potřeba vytvářet animaci podle zadaných hodnot. Bohužel toto render neumožňuje. Render funguje na principu, že nám ze zadaných hodnot vytvoří snímek nebo celou animaci. My místo toho využijeme Blender GameEngine , který nám umožňuje spouštět animaci přímo v Blenderu.

Jako poslední možností v horní liště je výběr prostředí (momentálně využíváme prostředí modelovací (Obr.1.1)) . Zvolit si můžeme hned z několika možností a to Animačního, kde se pracuje s IPO křivkami, Materiálového, které nám usnadňuje práci s barvami a texturováním objektu, Sequencer pro práci s videem a jako poslední Scripting, které je přizpůsobené pro práci s Pythonem.

1.2.2 3D okno

Orientace v 3D prostředí Blenderu je z počátku trochu nepřehledná, ale po delším používání zjistíte, že je celý Blender koncipován pro rychlou a jednoduchou práci. Většina funkcí, které potřebujeme, se dá vyvolat několika způsoby. První možnost je časově delší a to pomocí nabídek v Blenderu.[2] Lepší možnost je využití klávesových zkratek. Například rozdělení plochy na několik menších částí lze udělat s následujícím postupem. V dolním menu 3D okna vybereme menu Mesh->Hrany->Rozdělit , tím dojde k rozdělení plochy na 4 menší. Další možnost je pomocí klávesové zkratky „W“ a z menu vybrat možnost Rozdělit.

1. 2. 3. 4a-b. 5. 6. 7. 8. 9.

Obr. 1.3 Menu 3D okna

Popis menu v 3D okně (Obr.1.3)

1. umožňuje změnit typ celého okna, například na uživatelské nastavení viz. Horní lišta nebo ovládací panely. Je zde možné vybrat z mnoha možností, které nám zpřístupňují různá nastavení Blenderu

2. v položce Pohled jsou možnosti pro výběr kamer, pohledu, dále výběru pozadí na 3D plochu, které nám může pomoci při modelování složitějšího objektu

3. pokud máme na scéně více objektů, může být časem složitější se v nich orientovat. V tomto nám pomůže menu Vybrat, kde máme možnost zvolit seskupené objekty, objekty se stejnými vlastnostmi nebo stejného druhu 4a.

4.A menu Objekt je asi nejdůležitější. Nachází se zde možnosti pro modelování a úpravu objektu, proto si projdeme toto menu podrobněji

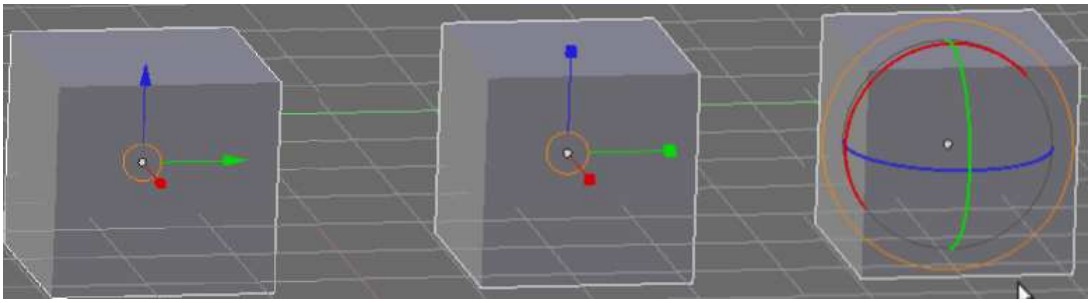
- **skripty** -zde můžeme vyvolat skripty související s modelováním
- **Show/Hide Objects** – slouží k dočasnému schování objektu na scéně. Pokud budeme mít velkou rozpracovanou scénu, projeví se to hlavně na zátěži počítače a tím i zpomalení naší práce. Proto je lepší nechat viditelné pouze objekty, se kterými právě pracujeme a zbytek pomocí nabídky schovat.
- **přesunout do vrstvy** - podobná možnost jako Show/Hide je možnost přesunout do vrstvy. Tato funkce se nejvíce používá k rozdělení scény například na objekty a světla nebo případně jinak
- **změnit typ objektu** - využijme pokud je potřeba převést křivky na mesh. [2]
- **spojit objekty** – použijeme pokud potřebujeme spojit několik objektů do jednoho, usnadňuje to práci s deformacemi . [2]
- **Boolean operace** nám umožní základní matematické operace s objekty, sčítání, odčítání[2]
- **Klonovat, duplikovat** – rozdíl je zde takový, že pokud těleso naklonujeme, stává se obrazem původního tělesa. Pokud změníme na původním velikost, barvu atd., změní se i klonované těleso. Při použití funkce duplikovat se vytvoří stejná kopie jako původní těleso, ale nadále nezávislá na vzoru. [2]

4.B pokud se přepneme do editačního menu (viz. Editační mód), dostáváme se k možnosti editovat body, křivky nebo plochy. Menu objekt se změní na Mesh a možnosti tu jsou trochu jiné.

- **Plošky** – máme možnost vytvářet nové plochy, převádět čtyřúhelníky na trojúhelníky a naopak. [2]
- **Hrany** – zde máme možnost rozdělit jednu plošku na několik malých, zaoblit hrany mezi více ploškami [2]
- **Tažení** – pokud označíme například 1 plochu a dáme tažení, zachová se původní tvar objektu a k tomu se vytáhne z předešlého objektu vybraná ploška. Tato metoda se nejčastěji používá při modelování. [2]
- **Editační mody** – zde máme na výběr z několika možností a tím se nám i mění možnosti práce s objektem. Mezi nejdůležitější patří Objektový mód, kdy můžeme tělesu měnit polohu, natočení a celkové velikosti, ale nemáme přístup k jednotlivým částem objektu. Od toho máme Editační mód, kdy můžeme ve vybraném objektu měnit pozice Body, Hran i Ploch a dále s nimi pracovat. Zbylé módy nejsou pro nás momentálně důležité. [2]

5. Zde můžeme vybrat z 5-ti zobrazení, jaké bude mít 3D plocha, jsou to :

- **Obálka**- zobrazí se pouze největší velikost objektu[2]
- **Drát**- zobrazí se celé těleso, kde jsou vidět všechny body a hrany[2]
- **Plošný**- jsou vidět celé objekty, ale používá se osvětlení z našeho pohledu [2]
- **Stínovaný**- podobný jako plošky, ale zde se používá osvětlení podle světel, které máme na ploše. [2]
- **Texturovaný**- pokud je tělesu přiřazena textura, tak by nyní měla být viditelná. Pokud není, je potřeba zkontrolovat zda normály tělesa směřují směrem ven a případně je přepočítat(Ctrl + N) [2]



Obr.1.4 Možnost změny pozice, velikosti a rotace pomocí kurzoru

6. udává podle čeho se bude rotovat objekt. Pokud zvolíme Medián, bude objekt rotovat kolem svého středu. Jiná možnost je použití 3D kurzoru a umístění ho mimo těleso. Těleso bude nyní rotovat kolem 3D kurzoru.

7. Ikona ruky nám udává, jaké možnosti práce s objektem budeme moci provádět v 3D okně přímo pomocí myši. Možnosti jsou v následujícím pořadí (Obr.1.4).

   - **pozice**- umožní měnit pozici kliknutím na 1 z os X,Y,Z [2]
   - **rotace**- umožní rotaci objektu kolem os X,Y,Z [2]
   - **zvětšení/zmenšení**- umožní měnit velikost objektu na osách X,Y,Z [2]

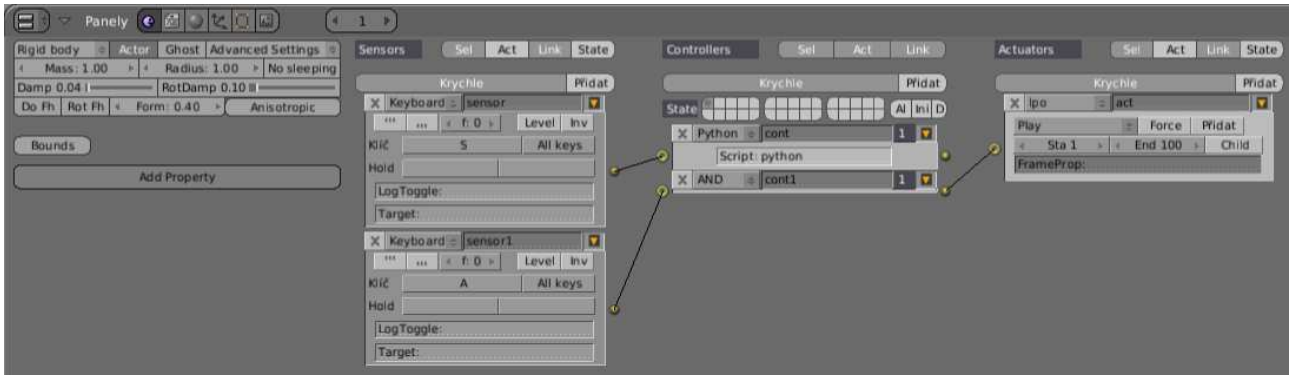
   Tyto možnosti jdou provádět i číselně a to pomocí menu Vlastnosti transformace, které vyvoláme klávesou N.

8. zde je možnost výběru vrstev, s jakými chceme pracovat. Pokud potřebujeme vybrat více vrstev, stačí použít klávesu Shift a pak vybrat jaké vrstvy potřebujeme.

1.2.3 Ovládací panely

Tento blok patří mezi nejdůležitější. Nastavují se zde vlastnosti těles, barvy, dynamika, nastavení kamer a světel i prostředí. Projdeme si zde pouze důležité základy, které budeme využívat později při vytváření animace. Popíšeme si, jak přiřadit objektu barvu, přiřadíme jednoduchou texturu, nastavíme si jednoduchá světla a ukážeme si základ GameEngine(logiky).

Začneme od začátku. Jako první možnost v Ovládacích panelech je Logika.



Obr.1.5 Zobrazení nastavení logiky v ovládacích panelech

1.2.3.1 Logika

Můžeme si jí rozdělit na 4 části (Obr.1.5). V první zleva máme možnost nastavovat vlastnosti těles. V prvním rolovacím menu máme na výběr několik možností, které si nyní popíšeme :

*a)**soft body*** – pomoci tohoto nastavení můžeme simulovat měkká tělesa, například dopadající míč na tvrdou podložku. Míč se při dopadu na podložku podle nastavených hodnot zdeformuje a odskočí znovu nahoru. Pokud je správně nastavené prostředí, měl by se míč chovat jako při reálné situaci [2]

*b)**Ridic body*** – zde se řeší kolize těles. Můžeme tak například simulovat hru kuželky. Budeme potřebovat pouze dráhu pro kuželky, kterou nám může nahradit obyčejná Plocha. Dále budeme potřebovat několik kuželek, které nahradí několik kvádrů a jednu kouli. Objektům nastavíme Ridic body. Dále musí nastavit velikost pole okolo každého objektu, kde se má kolize počítat. To lze udělat buď nastavením položky Radius, aby odpovídala velikosti objektu . Druhá možnost je trochu jednodušší nastavením Bounds na objekt jaký máme zvolený. V našem případě zvolíme pro kulečníkovou kouli Share a pro kuželky Box. Nyní pokud spustíme animaci (ukazatel myši musí být na 3D okně a dále zmačkneme klávesu P zjistíme, že koule pouze kuželky vychýlí, ale neporazí je. Důvodem je nastavení stejné váhy kuželky i koule. Toto můžeme lehce napravit, pokud nastavíme Mass na větší hodnotu než 1. Čím větší bude váha, tím větší by měla mít koule razanci. Podobnou metodu využijeme při animaci šikmého vrhu, kdy dělová koule řízená výpočtem balistické dráhy narazí do stavení sestaveného z objektu s nastavenými Ridic body. [2]

Zbylé tři nabídky slouží k ovládání logiky. První nám udává na jaký podmět se má něco udělat, druhá jaká logická funkce má být použita (AND,OR, XOR... nebo skript python, kde můžeme nastavit, co se má dít a poslední,co se má provést.

Pokus si prohlédneme podrobněji obrázek, zjistíme, že objekt Krychle má nastavené dva aktivátory. První se spustí, pokud zmáčkneme klávesu S, tím se spustí skript python, kde můžeme definovat, co se má dít. Můžeme zjistit, zda se naplnily nějaké podmínky, nebo jako v našem případě načteme hodnoty z textového souboru a pomocí skriptovacího jazyka vypočítáme balistickou křivku a přiřadíme ji objektu.

Ve druhém spouštěči čekáme na stisknutí klávesy A. Poté se projde funkcí AND, kde se nám odstartuje animace definovaná pomocí IPO křivky, kterou jsme spočítaly při stisknutí klávesy S.

Podmětů na spuštění může být několik. Můžeme spouštět například z klávesnice jako z ukázky nebo stisknutím myši. Další pro nás zajímavá možnost je Always, která se spustí pří spuštění GameEngine (klávesa P) a nečeká až se například zmáčkne klávesa. Ostatní možnosti nebudeme využívat.

Podobně jako je tomu u podmětu, máme na výběr i u akce, která se má provést. V ukázce spouštíme pohyb objektu po dané dráze, ale objekt můžeme například jenom posunou po některé ose o zvolený krok. Dále můžeme nastavovat kamery, scény a další nastavení .

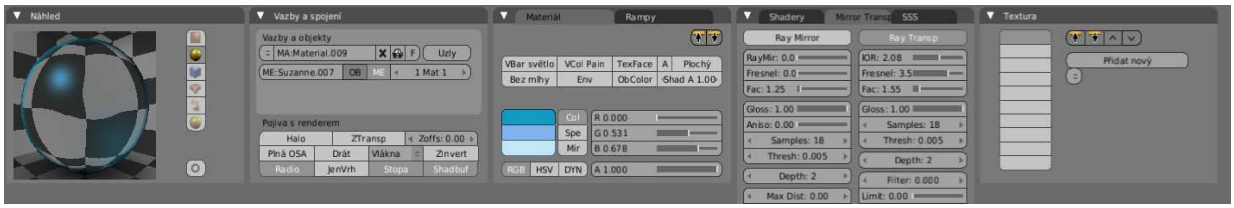
1.2.3.2 Stínování

ve stínování najdeme nástroje pro nastavení světel, barev i textur. Projdeme si postupně jednotlivé podložky, které budeme využívat.

Obr.1.6 Zobrazení nastavení světla v ovládacích panelech

a)***světla*** – pokud vytvoříme v Blenderu nějakou scénu, musíme ji i vidět. K tomu potřebují objekty světlo. Máme jich několik druhů. Paří mezi ně i Světlo a Slunce. Zbylé nebudeme využívat, protože potřebujeme, aby scéna byla co nejjednodušší. Světla jsou důležitá při renderování snímku nebo videa. My bude pracovat s přímo přehrávaným videem, a proto by nebylo výhodné čekat na 1 snímek několik sekund, když potřebujeme alespoň 25 snímků za sekundu, aby bylo video plynulé.

Světlu můžeme nastavit několik parametrů. Záleží jaký druh světla právě využíváme. My se zaměříme na Světlo. V položce energy (Obr.1.6) nastavujeme sílu světla. Zde si musíme dát pozor, aby scéna nebyla příliš tmavá, ale také na to, abychom scénu nepřesvětlili. Další možností je nastavení barvy světla. Můžeme tak například simulovat slunce, nebo modré neonové světlo. [2]



Obr.1.7 Zobrazení základního nastavení světla v ovládacích panelech

Obr.1.8 Zobrazení základních možností nastavení barvy objektu

Jako další můžeme nastavovat stíny, kde podle nastavené kvality může render jednoho obrázku trvat několik sekund až desítek minut, pří složitějších scénách i několik hodin.

b) ***barvy*** – zde máme možnost nastavovat velký rozsah barev pomocí RGB parametru. Pokud nějaký parametr změníme, ukáže se nám výsledek v levém okně(Obr.1.7), kde můžeme zvolit, na jakém objektu by měla být vidět ukázka. Spolu s ukázkou se změní i základní barva objektu, ale složitější věci, jako například průhlednost, odraz a textury, se nám projeví až při renderu.

V Pod menu Vazby a spojení(Obr1.7) přidělujeme jednotlivé barvy objektům. Je možné, aby 1 barvu mělo více rozdílných objektů. Dále tu máme nastavení, co z daného objektu bude vidět, jde nastavit, aby objekt nevrhal stín nebo použít průhlednost. [2]

V položce Materiál(Obr.1.7) nastavujeme barvy pomoci RGB parametrů. Zde se ještě dělí nastavení na 3 části Col, kde nastavujeme barvu objektu, Spe, zde nastavujeme jakou barvu bude mít průhledný objekt a nakonec Mir, kde se

nastavuje barva odrazu.

Další skupina se zabývá odrazem světla, odleskem a průhledností a nakonec novou funkci v Blenderu a to je simulace průsvitnosti. Shadery udávají, jak se má chovat odraz světla od objektu. Jde nastavit, aby Objekt vypadal jako plastový, kovový nebo i bez odrazu světla. [2]

Mirror Transp nám nastavuje, jestli bude objekt průhledný, lesklý nebo obě vlastnosti najednou.

Jako poslední je SSS(Subsurface Scattering), kde se simuluje průsvitnost objektu. Dá se to využít pro prosvícení pokožky ruky, kde slabší části kůže budou světlejší. Stejného efektu dosáhneme v reálné situaci, pokud si ruku prosvítíme baterkou. [2]

Na Obr.1.8 je ukázáno několik příkladových možností, jak lze barvy nastavit. Na prvním obrázku je pouze obyčejná barva. U druhého je nastaven Halo efekt, kdy se každý bod v objektu změní na zářící. Třetí a čtvrtý objekt mají nastavené různé barevné přechody. U pátého je použit odraz od okolí, kdy odráží částečně objekty v jeho okolí a barvu prostředí, ve kterém je. Šestý objekt má nastavenou průhlednost bez lomu světla. Předposlední objekt má nastavený odraz s průhledností a byly přiděleny barvy pro průhlednost a odrazy. Poslední objekt využívá SSS, kde je vidět, že tenčí části objektu jsou světlejší.

Obr.*1.9 A)Plocha s použitím textury* *B) Plocha s použitím textury s Nor*

Obr.1.10 Nastavování vlastností textury v ovládacích panelech

c) ***texturování*** – pokud potřebujeme dát obrázku vzhled nějakého reálného objektu, budeme nejspíše potřebovat textury. Textury se dají přiřadit jednotlivým objektům. Na obrázku Obr.1.9A vidíme plochu, na kterou byla textura nebe. U obrázku Obr. 1.9 B byla použita funkce Nor, která dokáže pomocí bump maps nasimulovat prostorový povrch. Toto se dá využít ke zmenšení počtu bodů objektu. Například můžeme vymodelovat lidský obličej, který může mít ve finálním stádiu i několik milionů bodů. Pokud bychom takovýto obličej použili ve hře, moc bychom si nezahráli. Proto využijeme bump maps, kdy z našeho obličeje použijeme UV mapu. V reálu si to můžeme představit, jako kdyby jste použili z hlavy kůži a tu pak natáhli na podobný model hlavy, který bude obsahovat pouze pár tisíc bodů. Pokud použijeme pouze UV mapu, budou na modelu vidět vrásky, ale nebude to vypadat jako na původní hlavě. S využitím Nor se vrásky nasimulují jako by byly vymodelovány prostorově. [2]

1.3 Internet

Internet v dnešní době zastává neocenitelnou úlohu v životě člověka. Šetří náš čas i finance, pomáhá nám nejen při nakupování, ale také při využití volného času či hledání zaměstnání. Je to také důležitá forma komunikace. Pomocí internetu docílíme rychlé a efektivní vyhledání informací.

1.3.1 Historie internetu

První náznak internetu vznikl v roce 1968 ve Velké Británii[1]. Zde se jednalo o základní sít, která neopustila budovu. Do vývoje internetu se pustila americká armáda, která chtěla zajistit komunikaci a to i v případě, že by došlo v sítí k poruše. V RAND

Corporation přišli na řešení, jak tento problém vyřešit. Využili síť bez centrálního uzlu, kde se při výpadku jedné trasy zvolí jiná a tím zůstane zachováno spojení. Vládou USA byla založena organizace Advanced Research Projects Agency, která v roce 1969 vybudovala experimentální síť ARPANET. Zpočátku nebyla síť moc využívána, byly připojeny státní úřady a některé university. [1]

Zlom nastal, když Tim Berners-Lee začal v roce 1989 zkoumat práci s hypertextovými dokumenty. Z této práce později vzniklo World Wide Web, kde je možné přenášet texty i obrázky. V roce 1992 bylo k internetu připojeno již přes milion uživatelů. Internet začali využívat i pro komerční účely.[1]

1.3.2 Vznik HTML

Nyní již víme jak vznikl internet ale jak je možné, že připojením se na některou internetovou stránku se nám zobrazí texty, obrázky, video či hudba.

Mezi základní způsoby patří HTML (HyperText Markup Language) V překladu hypertextový (hypertext = odkaz) značkovací jazyk.

Podle [3] vzniklo HTML na základě Tim Berners-Lee a Robert Caillau, kteří pracovali na propojeném informačním systému pro CERN ve Švýcarsku. Pracovali v programech TeX, Postskript ale potřebovali jednoduší jazyk. Proto v roce 1990 vytvořen jazyk HTML a následně protokol pro přenos dat v počítačové síti http(HyperText Transfer Protocol)

1.4 Python

Jazyk Python vznikl v roce 1989 ve výzkumném ústavu v Amsterodamu [7]. Na jeho vývoji se podílel Guido van Rossum. Jazyk Python dostal jméno podle pořadu BBC Monty Python's Flying Circus.[7]. Dnes se program rozšířen na mnoha operačních systémech (Windows, Linux…)

Python je objektový jazyk, kde jsou řešeny výjimky (pokud nastane chyba v části programu a máme ošetřenou výjimku, zpracuje se jiná část programu.

Do Pythonu jdou implementovat věci z programovacího jazyka C++, ale lze to provést i opačně. Základním rozdílem v jazyku C++ a Python je jiné řešení částí kódů za podmínkou.

Pokud používáme programovací jazyk C++ stačí část kódu umístit do {} u Pythonu musíme část kódu, která patří k podmínce odsadit.

1.5 Fyzika

Při popisu jednotlivých vrhů se vychází ze stejných rovnic, které jsou pouze upravené. Pro popis vrhů použijme šikmý vrh. Zbylé jevy se dají nasimulovat změnou parametrů. Pokud bude nastaven úhel šikmého vrhu na 90°, bude se jednat o děj vrh vzhůru.

1.5.1 Šikmý vrh

Se šikmým vrhem se setkáváme již od vzniku lidstva. Lidé v pravěku využívali šikmého vrhu k lovu zvířat. I když si sice neuvědomovali, podle jakých zákonů se letící objekt pohybuje, dokázali ho využít.

1.5.1.1Matematický popis

Šikmý vrh můžeme rozdělit na dvě částí. Na vodorovný a svislý pohyb, přitom jsou oba pohyby na sobě nezávislé [6]

$$x - x_0 = v_0 + \frac{1}{2} a_x t \quad [6] \qquad (1)$$

a) Vodorovný směr - víme, že vodorovná složka tíhového zrychlení je nulová, proto na čase nezávislá [6]. Těleso si udržuje počáteční rychlost $v_{0x,}$ a proto posunutí tělesa ve vodorovném směru $x\text{-}x_0$ vypočítáme podle vzorce(1) pr0 $a_x=0$

$$x\text{-}x_0=v_{0x}t \quad [6] \qquad (2)$$

$$v_{0x}t= v_0 \cos\Theta_0 \quad [6] \qquad (3)$$

Když dosadíme vzorec (3) do vzorce (2) získáme:

$$x\text{-}x_0 =(v_0 \cos\Theta_0)t \quad [6] \qquad (4)$$

b) Svislý směr- svislou složku vektoru posunutí dostaneme ze vzorce :

$$y - y_0 = v_{0y}t - \frac{1}{2}gt^2 = (v_0 \sin\theta_0)t - \frac{1}{2}gt^2 .[6] \quad (5)$$

Svislou složku rychlosti v_{0y} můžeme nahradit $v_0 \cos\Theta_0$ a následně upravit na:

$$v_y = v_0 \sin\theta_0 - gt \qquad [6] \qquad (6)$$

$$v_y^{\,2} = (v_0 \sin\theta_0)^2 - 2g(y - y_0)\,[6] \qquad (7)$$

1.5.1.2Výpočet šikmého vrhu bez vlivu prostředí

Při šikmém vrhu máme těleso v počátečním stavu v počátku kartézských souřadnic, pak je v čase $t=0$ $x_0 = y_0 = z_0 = 0$

$v_{0x} = v_0 \cos\alpha$ $\qquad$ $v_{0y} = v_0 \sin\alpha$ [5] $\qquad$ (8)

Počáteční rychlost spočítáme pomocí vzorce (8)

$v_x = v_0 \cos\alpha$ $\qquad$ $v_y = v_0 \sin\alpha - gt$ *[5]* $\qquad$ (9)

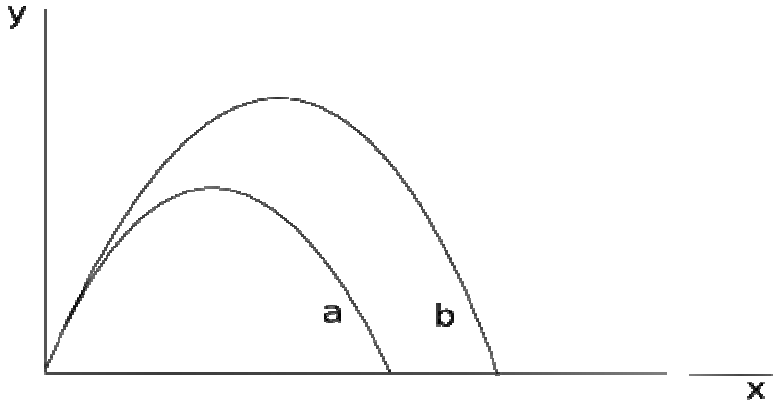
Pokud potřebujeme znát rychlost tělesa v čase t>0 využijeme vzorce (9)

$$x = v_0 t \cos\alpha$$

$$y = v_0 t \sin\alpha - \frac{1}{2} g t^2 \qquad [5] \qquad (10)$$

Při výpočtu pozice tělesa v čase použijeme vzorec (10)

1.5.1.3 Výpočet šikmého vrhu s vlivem prostředí



Obr.1.11 Ukázka změny trajektorie tělesa a)pohybujícího se ve vzduchu b)pohybujícího se ve vakuu

Až doposud jsme uvažovali, že na těleso nemá vliv okolní prostředí. Pokud ho chceme započítat, musíme přidat několik parametrů. Před výpočtem musíme mít zadané hodnoty hmotnosti, poloměru tělesa a hustotu prostředí, ve kterém se těleso bude pohybovat. Pokud se těleso pohybuje v hustém prostředí, jeho trajektorie bude odlišná než u trajektorie tělesa pohybujícího se ve vakuu Obr.1.11

$$S = 4\pi R^2 \qquad [5] \qquad (11)$$

$$k = C\frac{\rho}{2}S \qquad [5] \qquad (12)$$

Pomocí vzorce (11) získáme velikost povrchu koule. Následně spočítáme koeficient odporu prostředí k (12) .

$$F_y = -gm - kvv_y \qquad F_x = -kvv_x \qquad [5] \qquad (13)$$

$$a_y = F_y/m \qquad a_x = F_x/m \qquad [5] \qquad (14)$$

$$v_y = v_y + a_y dt \qquad v_x = v_x + a_x dt \qquad [5] \qquad (15)$$

$$y = y + v_y dt \qquad x = x + v_x dt \qquad [5] \qquad (16)$$

Nyní spočítáme sílu (13), z ní následně zrychlení tělesa (14). Z vypočítaných hodnot získáme rychlost na osách X a Y (15) a nakonec pozici tělesa v daném čase (16).

2 Praktická část

V praktické části se budou vytvářet jednotlivé části projektu. Základ bude tvořen HTML stránkou, pomocí které zobrazíme informace o daném ději a budou vytvořeny jednotlivé části pro grafický vzhled internetové stránky. Poslední částí je vytvoření pomocí programu Blender a s pomocí programovacího jazyka Python animace. Vytváření vstupních dat a simulace trajektorie bude řešit program, který není součásti této práce.

2.1 Tvorba html stránek

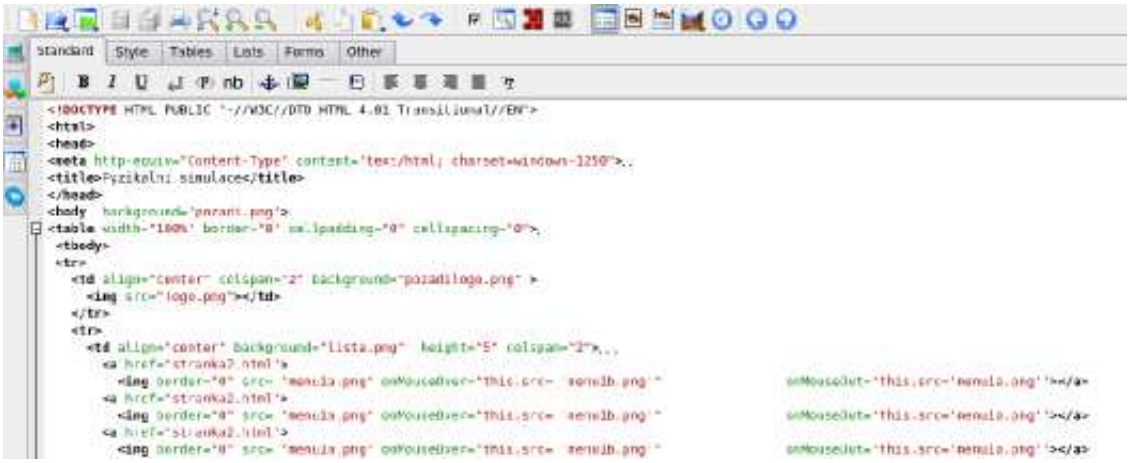
Internetové stránky pomáhají předávat informace o tom, co nás zajímá nebo co se

potřebujeme dozvědět. Mezi několik způsobů, jak internetové stránky vytvořit patří i metoda HTML. Takovéto stánky můžeme umístit na internet několika způsoby. Mezi ně patří služby free web-hostin, kde jsme omezeni velikostí datového úložiště pro stránky a na takovéto stránky se musí přidat reklama, ale za umístění se nic neplatí. Další možnost je web-hosting, kde můžeme dle dohodnuté ceny používat určité služby. Pokud je potřeba vytvořit internetové stánky, kde je nutné počítat s velkým přístupem na tyto stánky, nebo jsou požadavky na kvalitní správu stránek, jde použit server hostin.

HTML kód je tvořen pomocí tagů(značek) [4]. Například, pokud budeme chtít, aby se na stránkách zobrazil tučný text, můžeme to udělat takto: „tučný text“ výsledek bude po zobrazení v některém z dnešních prohlížečů vypadat takto „**tučný text“** . Dosáhneme toho tím, že text umístíme to tagů. První znamená, že od tohoto bodu bude text tučný. Tučný text ukončíme pomocí . Podobně můžeme udělat například kurzívu. Stačí, abychom místo dali . Nesmíme zapomenou nakonec použít abychom kurzívu ukončili.

Pokud budeme psát nějaký text, zjistíme, že píšeme stále na jednu řádku a na další se dostane až po zaplnění celého řádku. Toto se dá napravit pomocí tagu <br>. Zatímco tagy pro tučný text a kurzívu jsou párové (musí začít a někde se ukončit) u tagu <br> to tak není. Tag <br> nám odřádkuje na další řádek a tím ušetří problémy s úpravou. [4]

2.1.1 Vlastní tvorba HTML stránky



Obr.2.1 Ukázka zobrazení HTML kódu v editoru Quanta Plus

K vytvoření HTML kódu je výhodné používat HTML editory. Editory nám umožňují lépe se orientovat v textu, zobrazují chyby, kde je špatný kus textu, ale některé umí i vytvářet HTML stánku bez nutnosti pokročilé znalostí psaní kódu v HTML. Pro tvorbu stánek byl použit editor Quanta Plus Obr.2.1 . Nyní si popíšeme jednotlivé části kódu.

<html tag html nám udává začátek stránky

<head> head nám udává podrobnější informace o stránce

<meta http-equiv="Content-Type" content="text/html; charset=windows-1250"> nastavujeme zde, že se jedná o HTML stránky a použití české znaková řady windows-1250

<title>

</head> konec hlavičky

<body> zde začíná tělo stránky, vkládáme texty,obrázky..

<body background="pozadi.png"> definujeme, že celá stránka má mít jako základní pozadí použito „pozadi.png".

<table width="100%" border="0" cellpadding="0" cellspacing="0"> - nastavujeme tabulku, kde parametr width udává velikost stránky v procentech. Je možné používat i pixely, ale při změně rozlišení stránky by byly tabulky posunuté. Takto docílíme toho, že tabulka bude vždy přes celé okno. Ostatní parametry nám udávají velikost rámečku a vnitřní a vnější velikost rámečku tabulky. My potřebujeme, aby tabulka nebyla vidět. To snadno docílíme nastavením parametrů na nulu.

<tbody>,<tr>,<td> - tbody nám udává začátek těla tabulky. Tag tr nám udává řádek tabulky. Počet řádků není na začátku nastaven, to znamená, že pokud potřebujeme nový řádek, přidáme do kódu párový tag <tr></tr>. Pokud potřebujeme novou buňku, použijeme párový tag <td></td>

<td align="center" colspan="2" background="pozadilogo.png" > - tím docílíme, aby zarovnání v buňce bylo na středu, pomocí colspan=2 nastavíme, aby tato a další buňka byla spojena do jedné. Pomocí background dáme na pozadí buňky obrázek .

****

<img border="0" src= „menu1a.png" onMouseOver="this.src= ‚menu1b.png'"

onMouseOut=“this.src=‘menu1a.png‘“>

****

pomocí párového tagu <a> říkáme, že se po zmáčknutí má zobrazit stránka „stranka2.html“ . Uvnitř tagu <a> bude tag na umístění obrázku. Výsledek bude vypadat tak, že na začátku bude základní obrázek. Pokud na obrázek najedeme myší, obrázek se změní na jiný. Pokud budeme na obrázku a zmáčkneme myš, přesměrujeme se na novou stánku. Tag IMG nám definuje, že teď se nastaví obrázek. Pomocí border nastavíme, aby obrázek neměl žádný rámeček a příkazy onMouseOver a onMouseOut nastavíme změnu obrázku při najetí myši na obrázek a při dojetí z obrázku. Základní obrázek nastavíme na začátku pomocí SCR.

</tbody> ukončuje tělo rámečku

</table> ukončuje rámeček

</body> ukončuje tělo HTML

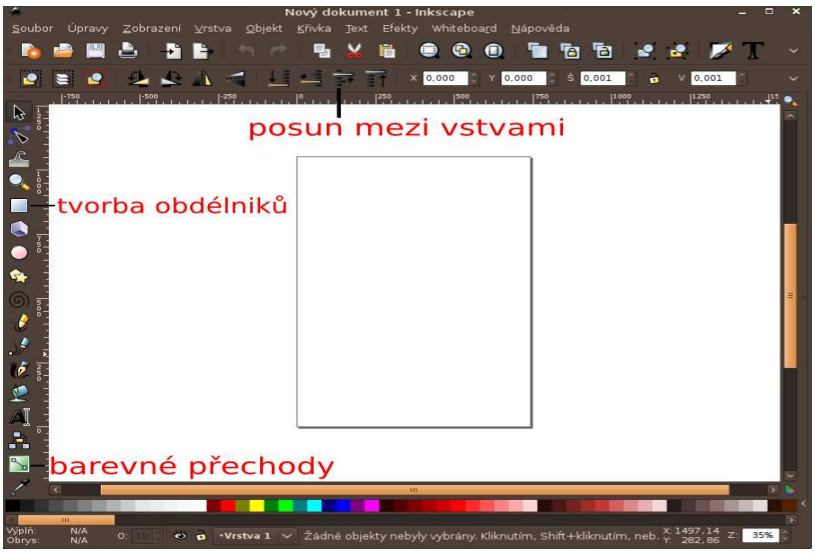
</head> ukončuje HTML stránku

2.1.2 Tvorba vzhledu HTML stránek

Obr.2.2 Hotová HTML stránka

Samotný kód většinou nestačí na vzhled stránek. Jde sice nastavit barva pozadí, měnit barvu písma a podobné maličkosti. Pokud potřebujeme vylepšit vzhled stránky, použijeme obrázky, například na pozadí, na nadpisy i na tvorbu ikon.

Obrázky si můžeme stáhnout z internetu a to buď volně dostupné, nebo si více kvalitní koupit. My zvolíme třetí variantu a to tu, že si je vytvoříme. Na internetu sice najdeme plno vhodných obrázků a tlačítek, ale dají se jen málo upravovat.



Obr.2.3 Rozhraní programu Inkscape

Pro tvorbu obrázků a tlačítek jsem zvolil program Inkscape. Jedná se o OpenSource program, to znamená, že je volně dostupný a není potřeba platit za drahé komerční programy. Program Inkscape je vektorový, to znamená, že objektům vytvořeným v Inkscapu můžeme měnit velikost bez ztráty kvality. Pokud bychom něco podobného udělali z obyčejným obrázkem, tak výsledek nebude vypadat dobře, budou vidět jednotlivé pixely a ostrost obrázku bude špatná.

Na tvorbu stránek budeme potřebovat dva obrázky na pozadí a čtyři tlačítka. Tlačítka budeme používat celkem dvoje, ale musíme mít další obrázky tlačítek, pokud na ně najedeme myší.

Začneme rychlým popisem Inkscapu. Nebudeme potřebovat mnoho funkcí, stačí nám pouze tvorba obdélníků, přechody a posun objektu (Obr.2.3) .

Tvorba pozadí, které bude na stránce, nám stačí vytvořit bílý obdélník, který nemusí být moc široký, ale musí být hodně dlouhý. Obdélníku dáme bílou barvu a umístíme na něj nový, stejně velký a dáme mu světle šedivou barvu. Nyní nám stačí jen umístit nad obdélníky barevný přechod a hodně ho natáhnout. Tím nám vzniklo první pozadí na stránky. Ze začátku je tmavší a postupně směrem dolů je světlejší.

Při tvorbě druhého pozadí budeme postupovat trochu jinak. Jako základ použijeme obrázek a upravíme ho do potřebné velikosti. Bude se jednat o obrázek, který bude

jako pozadí v nadpisu. Protože se jedná o pozadí, není přesná velikost nutná. Pokud bude stránka větší než obrázek, obrázek se částečně zopakuje.

Nyní vytvoříme tlačítka. Na tlačítka budeme potřebovat několik obdélníků. Prvnímu přiřadíme černou barvu. Obdélník zkopírujeme a dáme mu bílou barvu. Pomocí přechodu vytvoříme bílý pruh naspodu černého obdélníku. Bílý obdélník by měl být malý a rychle se vytratit dočerna. Znovu zkopírujme tentokrát bílý obdélník a otočíme ho o 180°. Tím dostaneme bílý pruh naho ře i dole. Výsledek si uložíme, protože ho budeme potřebovat na pozadí lišty s tlačítky. K dokončení tlačítek nám schází pouze přidání textu. Text se přidá v levé nabídce po zmačknutí ikony A . Dáme textu modrou barvu. Text si zkopírujeme a dáme stranou. Původní modrý text přidáme k černé liště s bílými pruhy. Protože chceme, aby tlačítko vypadalo, že je trochu prostorové, posuneme text o dvě úrovně níže, docílíme, že bílé pruhy ovlivní i text, který je pod nimi. Tím máme hotové 1. tlačítko, druhé uděláme stejným způsobem, jen použijeme jinou barvu textu. Dostaneme 2 stejná tlačítka, kde se liší pouze barva textu.

Jako poslední uděláme logo stránek. Vytvoříme text s názvem stránek a nastavíme mu červenou barvu. Za text přidáme logo Fyzikálního ústavu a máme hotovo. Výhoda tohoto uspořádání je, že obrázek je jenom text a logo. Zbytek je průhledný a neruší pozadí. Výsledný vzhled stránky je vidět na obrázku Obr.2.2

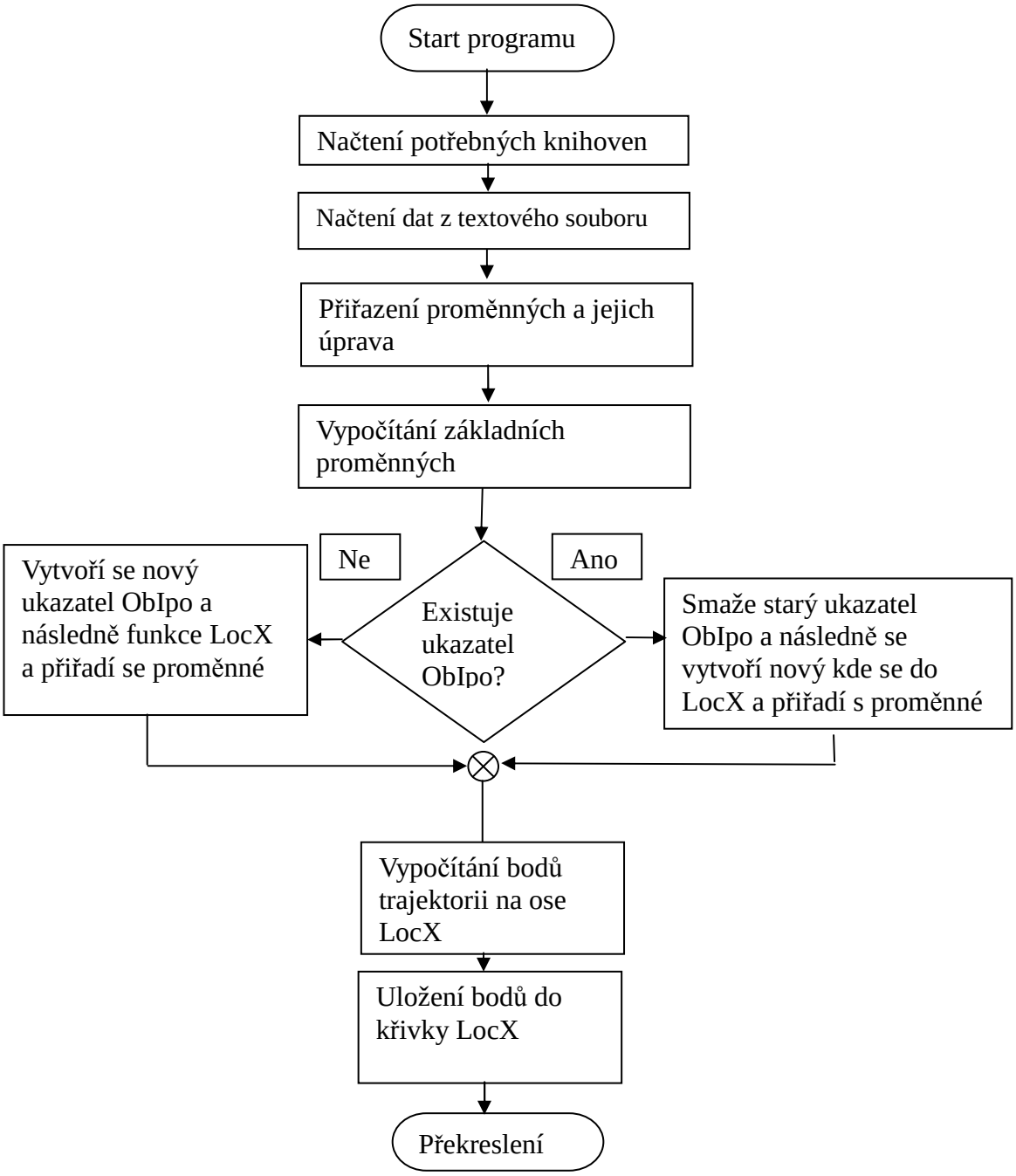
2.2 Python

Program Python je trochu podobný jazyku C++. Python se liší od C++ například při definování proměnných. V programovacím jazyku C++ je potřeba každé proměnné předem určit jaké hodnoty bude uchovávat (string, integer, float …), u jazyka Python na tom nezáleží [7] např. pokud uložíme do proměnné A například hodnotu 7.5, do proměnné B hodnotu 1.5 a A vydělíme B a to uložíme do proměnné C, zjistíme, že výsledná hodnota je integer, i když A i B jsou float. Program Python se tímto přístupem, jednoduchostí blíž ostatním programům jako jsou Matlab, PHP atd.

Jelikož projekt je koncipován tak, aby byl co nejuniverzálnější jsou podklady pro animace načítány z externího textového souboru, proto je nutné použít program Python externí výpočtový modul pro program Blender.

2.2.1 Vývojový diagram

Samotný Blender neumí pracovat s externími daty, proto je nutné načítat a pracovat s daty Pythonu. Vývojový diagram znázorňuje jádro celého výpočtu. Stejný postup se používá pro výpočet dalších os případně rotace. Ukazatel ObIpo nám spojuje objekt, se kterým pracujeme, s křivkami, které udávají cestu objektu v jednotlivých osách. Jméno ukazatele se může změnit, ale je nutné, aby se dále pracovalo se stejným názvem.



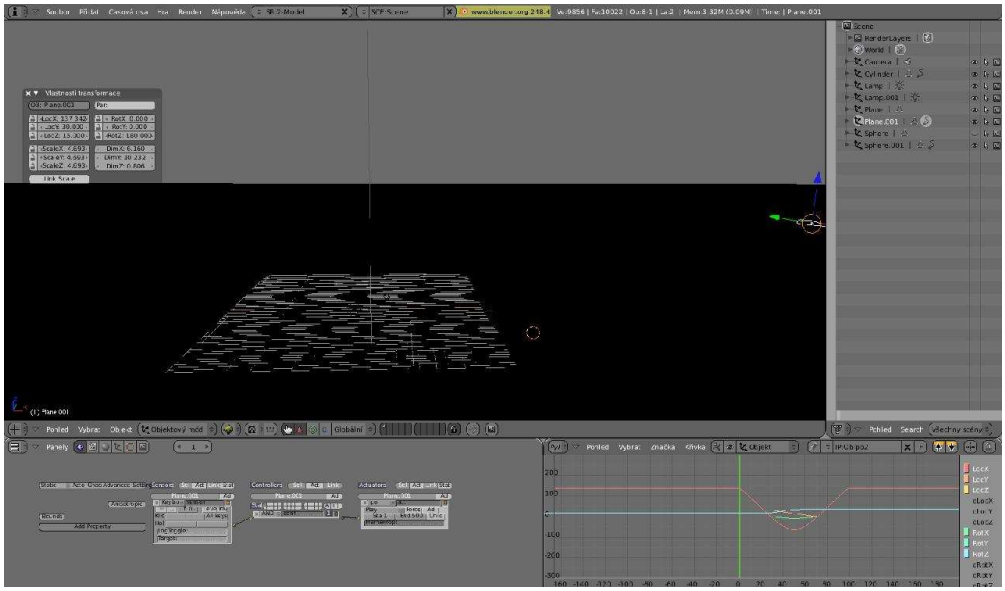
2.3 Tvorba programů

Animace zaměřené na praktické ukázání jevů jsou pouze částí celého bloku. Nejdříve bude student seznámen s teorii jevu a jeho matematickým popisem. Následně spustí animace fyzikální parametrů a na závěr si spustí animaci jevu v reálném prostředí.

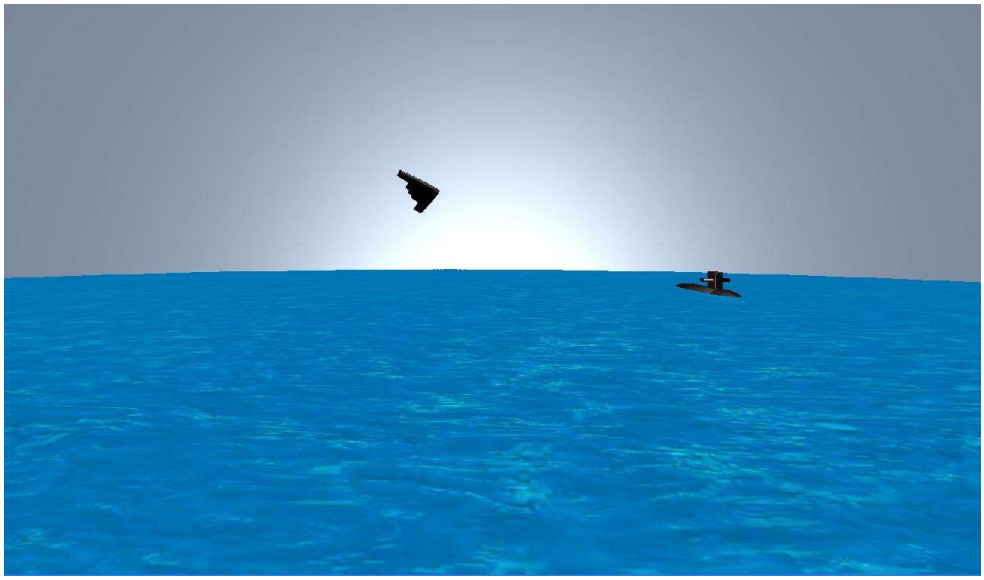
Programy vytvořené Blendrem nemají čitelný zdrojový kód, protože je využit celý program a v něm jsou upraveny jednotlivé části. Pro vytvoření programu musíme nejprve vymodelovat části scény, které budeme používat. Například budovy, přístroje, krajinu a další. Při vytváření jednotlivých objektů je třeba dbát na to, aby měly co nejméně bodů. Čím je scéna jednodušší, tím menší nároky jsou kladeny na zatížení počítače. Při vytvoření objektů byly použity postupy z kapitoly 1.3 . Následně potřebujeme zajistit výpočet trajektorie objektu. V tom nám poslouží Python skript. Program sestavený podle vývojového diagramu v kapitole 2.2.1 nám načte soubory z textového souboru, spočítá základní proměnné a následně spočítá samotnou trajektorii. Před výpočtem je možné vytvořit další natavení pro objekty, které ve scéně používáme. Například při animaci vodorovného vrhu na začátku určíme trajektorii letu letadla. To znamená, že podle načtených hodnot ze souboru nastavíme výšku letadla podle počáteční hodnoty *y0*. Po spočítání jednotlivých trajektorii je třeba jednotlivým objektům třeba dát vědět, že se mají řídit danou trajektorii.Toho docílíme tak, že objektu přiřadíme v zobrazení IPO křivek název ObIpo nebo jiný název, který jsme použily ve skriptu. Jako poslední část je potřeba objektům sdělit, kdy se mají spustit. To docílíme použitím Logiky Obr.1.5 . Tím určíme, že při zmáčknutí příslušné klávesy se provede výpočet ze skriptu a stisknutím dalšího tlačítka spustíme trajektorii. V animacích je to řešeno kombinacemi kláves P, O, I. Klávesou P spustíme rozhraní pro přehrávání her v našem případě prostředí pro zanimování děje. Klávesou O se provede spuštění Python skriptu, kde spočítá trajektorii objektu a pokud se na scéně vyskytují další pohyblivé objekty, nastaví jejich počáteční parametry. Poslední klávesa I spustí zanimování děje. Poslední částí je upravení vzhledu Blenderu. Jelikož se jedná o 3D grafický program, je zde i mnoho nabídek a možností nastavení. Lze to jednoduše upravit stisknutím pravého tlačítka myši na přechodu mezi jednotlivými okny.

2.3.1 Ovládání programu

Animace je přizpůsobená, aby byla co nejjednodušší na ovládání. Samotný program Blender je na ovládání velice složitý, a proto je vzhled programu upraven tak, aby uživatel, který nemá zájem pracovat s Blendrem mohl pouze pustit animaci a nebyl nijak zatěžován prostředím Blenderu. Jak je vidět podle obrázků 2.4 a 2.5 úprava zjednoduší celou animaci. Přestože je animace upravená, je stále možné do ní zasahovat klávesovými zkratkami. Kombinace kláves P, O, I je zvolena tak, aby bylo co nejpřehlednější. Klávesy O a I je možné zaměnit za jiné, ale klávesa P je nutná na spuštění potřebného rozhraní. Klávesy je možné měnit v Logice viz. Kap 1.2.3.1



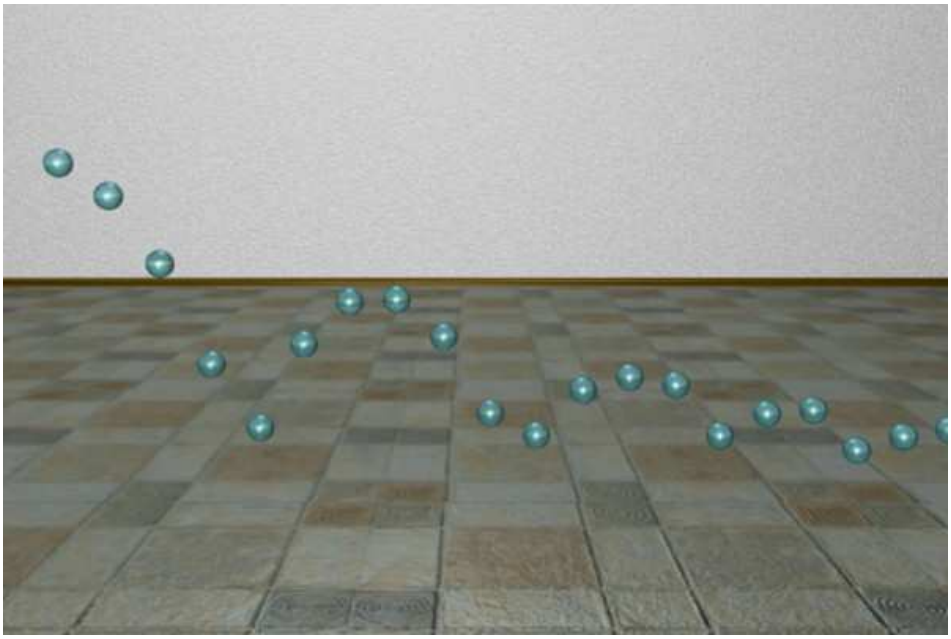
Obr.2.4 Vzhled animace před úpravou



Obr.2.5 Vzhled animace po pravě

2.3.2 Tvorba videa – vrhy

Při tvorbě videa se nám otevře velká paleta grafických nástrojů oproti GameEngine. Při vytváření animaci pro reálný průběh nemůžeme používat generování textur, osvětlení je značně omezeno. Je to způsobeno vysokou náročností těchto nástrojů, které nám neumožňují plynulý běh animací. Při vytváření videa se tímto ale řídit nemusíme. Předem si nastavíme, jak bude vypadat trajektorie objektu a můžeme využít i časové náročné procesy jako například výpočty stínů, různé odrazy a hlavně kvalitní nastavení materiálů. Například na Obr.2.6 je vidět použití Nor mapy na podlaze a použití odrazu na kuličkách. Jeden takovýto obrázek trvá vytvořit přibližně 10s. Pří 25 snímcích za sekundu by nám 4 sekundové video trvalo 17 minut. To pro video přehrávané v reálném čase nepřijatelné.



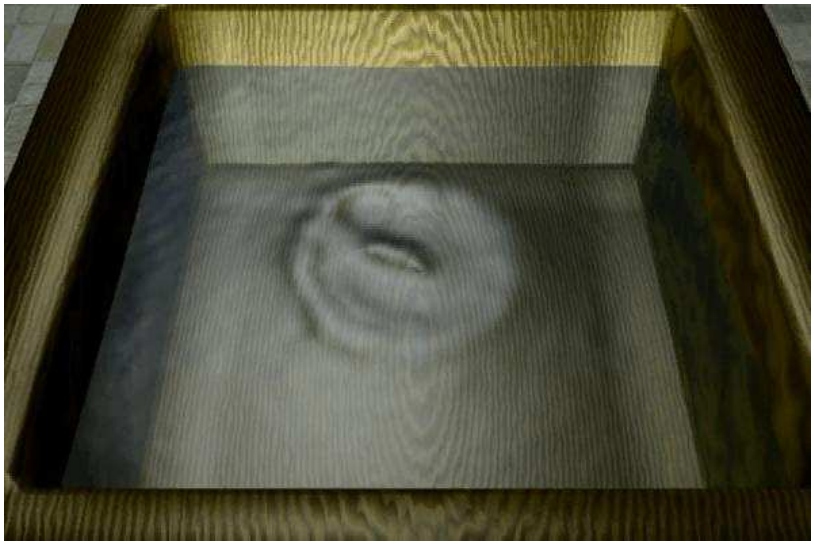
Obr.2.6 Zobrazení 1 snímku animace

Pro vytvořeni animaci nebyl použit Python skript, ale byla využita fyzika Blenderu. Fyziku jsme nemohli použít při vytváření předešlých animací z důvodu nepřesného nastavení základních hodnot. Nyní ukazujeme pouze průběh děje a počáteční hodnoty nezadáváme.

2.3.3 Tvorba animace – kapaliny

Blender umí simulovat kapaliny. Na ukázku, co se stane na vodní hladině, jsou vytvořeny 2 animace. První znázorňuje, co se stane, pokud na hladinu dopadne malá kulička. V druhé je kulička několikrát větší. Výpočet simulace kapalin je velice náročný

na čas. Pro vytvoření simulace je třeba nejprve připravit prostředí. Návod jak vytvořit prostředí je v kapitole 1.2. Pro vytvoření simulace potřebujeme 2 nové objekty nejlépe krychle. Krychle dáme nad sebe a podle toho, kolik chceme mít kapaliny, posuneme vrchní krychli dolů. Čím více se krychle překrývají, tím více budeme mít kapaliny. Následně je nutné nastavit vlastnosti těles. Na záložce Objekt-Fyzika-Fluid nastavíme pro vrchní krychli Domain a pro spodní Fluid. U objektu, který na kapalinu narazí nastavíme Obstacie. Pokud máme vše hotovo, klikneme na objekt, který má nastavenou vlastnost Domain na Bake. Tím se nám spustí výpočet kapaliny. Po výpočtu je nutné ještě vyrenderovat animaci. Toho docílíme tak, že zmáčkneme klávesovou zkratku ctrl+F12 . Výsledek vidíme na obrázku 2.7



Obr. 2.7 Jeden snímek z animace kapaliny (vlny)

Závěr

Cílem bakalářské práce bylo vytvořit nové prostředí pro simulaci fyzikálních jevů.Celek se skládá z několika částí. Na začátku jsou zpřístupněny informace pomocí internetových stránek. Následně je možné spustit program na simulaci fyzikálních jevů (tento program je náplní jiné bakalářské práce) a nakonec uvidí animaci jevů v reálném prostředí.

Původní myšlenka byla vytvořit pomocí programu Blender samostatný exe soubor, kde by byla animace daného jevu. Při realizaci tohoto postupu se vyskytl problém s knihovnami, které nebylo možné dodat. Řešením bylo použití celého programu Blender, kde je upravený vzhled, aby byl co nejjednodušší. K ovládání jednotlivých animací je potřeba 3 kláves a to P,O a I. U animací šikmého vrhu a vodorovného vrhu jsou použity scény, kde je kromě samotné animace i okolní prostředí. V případě vrhu vzhůru a volného pádu bylo okolní prostředí také použito, ale bylo příliš graficky náročné, a proto byla použita jen hlavní část animace. Jako doplnění animací byly přidány videa pohybu kuliček v reálném a ideálním prostředí a poslední část zobrazuje dopad tělesa do kapaliny. Kapaliny je možné vypočítat lépe a na větší plochy ,ale výpočetní čas by byl velice dlouhý v řádech hodin.

Celý systém bude v budoucnu sloužit jako výukový nástroj na ústavu Fyziky FEKT VUT v Brně. V budoucnu se předpokládá jeho rozšíření o další fyzikální jevy tak, aby systém obsahoval všechny potřebné tématické oblasti, které jsou na ústavu Fyziky probírány.

Použitá literatura

Internetové články

[1] BEDNÁŘ , Martin. Historie vzniku internetu. Historie vzniku internetu [online]. 2007 [cit. 2008-11-20]. Dostupný z WWW: <http://www.owebu.cz/internet/vypis.php?clanek=1083>.

Internetové stránky

[2]Blender Foundation. Www.blender.org [online]. 1998-2008 [cit. 2008-10-04]. Text v angličtině. Dostupný z WWW: <www.blender.org>

[3]HyperText Markup Language. Wikipedia [online]. 2008 [cit. 2008-12-13]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/HyperText_Markup

[4]JANOVSKÝ, Dušan. Jak psát web [online]. 2008 , 05. října 2008 [cit. 2008-11-11]. Dostupný z WWW: <http://www.jakpsatweb.cz/>

Skripta

[5]HRADILOVÁ, Eva,UHDEOVÁ, Naděžda. Fyzikální seminář. [s.l.] : [s.n.], 2003. 156 s.

[6] D. Halliday, R. Resnick, J. Walker: FYZIKA – český překlad Prometheus 2000

[7] ŠVEC, Jan. Učebnice jazyka Python (aneb Létající cirkus) Release 2.2 . [s.l.] : [s.n.], 2002. 90 s. Dostupný z WWW: <http://www.py.cz/UcebniceJazykaPython>.

Seznam zkratek

ARPANET - Advanced Research Projects Agency Network

CERN - European Organization for Nuclear Research

HTML - HyperText Markup Language

HTTP - HyperText Transfer Protocol

NaN - Not a Numer

SSS - Subsurface Scattering

WWW - World Wide Web

Přílohy

Na přiloženém CD-ROM jsou animace, zdrojové kódy, internetové stránky pro e-learning a bakalářská práce ve formátu pdf.

Animace jsou rozděleny na jednotlivé časti podle animovaného jevu. Uvnitř každé takovéto složky je soubor hodnoty.txt , který obsahuje počáteční hodnoty. Skript *.py, který obsahuje zdrojový kód pro výpočet trajektorie a soubor *.blend, který spustí animační prostředí. Před prvním otevřením souboru *.blend je potřeba nastavit spuštění pomocí Blenderu. Toho docílíme tak, že pravým tlačítkem myši klikneme na soubor *.blend , dáme Otevřít-> Vybrat program ze seznamu->Procházet a nyní na přiloženém cd najdeme složku blender-2.48a-windows a zvolíme program Blender. Dále se stačí řídit pokyny programu.

Na CD-ROM jsou dále ve složce WWW_stránky uloženy zdrojové kódy internetových stránek optimalizované pro prohlížeč Internet Explorer.